IN THE DRAWINGS:

BEST AVAILABLE COPY

Please note that formal drawings were filed in the U.S. Patent and Trademark Office on April 8, 2002. Submitted herewith are (2) sheets of formal drawings comprising Figures 1-2 and a copy of the postcard receipt.

10/060,750   2

REMARKS

Claims 2, and 8-34 are all the claims pending in the application. Claims 1, 3, 4, 5, 6, and 7 are canceled, above. Claims 2, and 8-34 stand rejected on prior art grounds. Applicants respectfully traverse these rejections based on the following discussion.

I.   The Prior Art Rejections

Claims 1-34 stand rejected under 35 U.S.C. §102(b) as being anticipated by Blaner, et al., "AN EMBEDDED PowerPC SOC for Test and Measurement Applications," 13th Annual IEEE International ASIC/SOC Conference, 2000, September 13-16, 2000, pages 204-208, hereinafter referred to as Blaner. Claims 1-34 stand rejected under 35 U.S.C. §102(e) as being anticipated by Devins, et al. (U.S. Patent No. 6,487,699), hereinafter referred to as Devins. Applicants respectfully traverse these rejections based on the following discussion.

The claimed invention provides a method, structure, and computer program product for a verification test bench system for testing an interface of a system-on-a-chip (SOC). As illustrated in FIG. 1 of Applicants' disclosure, the claimed invention comprises a SOC 100 connected to a verification test bench 300. More specifically, the verification test bench system includes a verification interface model 210 connected to the SOC interface 101 and a test bench external bus interface unit (EBIU) 200 connected to the verification interface model 210. The test bench EBIU 200 is connected to a SOC EBIU 205 within the SOC 100, wherein the SOC EBIU 205 allows a test case running in

10/060,750   11

the SOC 100 to control both the SOC interface 101 and the verification interface model 210.

Furthermore, the test bench EBIU 200 and the SOC EBIU 205 are mastered by the same processor in the SOC 100, such that the SOC interface 101 and the verification interface model 210 are programmed by the same test case running in the SOC. The test case is written to utilize software drivers to configure a core or units that the test case is testing.

Accordingly, one advantage achieved with the invention is better software control. The invention allows the test case executing in the SOC 100 to use the same software driver, if appropriate, to program both interfaces 101, 210. The test case can also use one driver to program the SOC interface 101 and another to program the interface model 210, both of which are controlled by the test case. This is an improvement over the conventional situation where the test case running within the SOC 100 controls the SOC interface 101, and another software program (written in a bus functional language) controls the external interface 210. The invention provides increased reusability and decreased development time because the invention uses the same or similar software written in the same language to program both interfaces.

In the rejection, the Office Action argues that Blaner discloses a memory-mapped external device containing software readable and writable registers that appear as wires in a testbench, which is connected to an external bus. In addition, the Office Action argues that Devins discloses external bus interface logic coupled to a SOC device via a chip-external bus. However, neither Blaner nor Devins disclose *connecting the testbench to*

10/060,750   12

an external SOC via the SOC interface and the model interface (independent claims 2,

8, 15, 21, and 28). Moreover, neither Blaner nor Devins disclose *a test case in the SOC*

that can use the same software driver to program both the SOC interface and the model

interface (dependent claims 4, 11, 17, 24, and 31).

Furthermore, the Office Action argues that Blaner expressly teaches connecting

the testbench to an external SOC via the SOC interface and the model interface; and, that

Devins expressly teaches an external memory-mapped test device (EMMTD) that is

coupled between a SOC design being tested in simulation, and cores external to the SOC

design. However, Blaner does not disclose the structural elements of an SOC interface

and a model interface, which connects the testbench to an external SOC. Moreover,

although Devins broadly discusses an SOC coupled to external components, Devins does

not teach or suggest connecting the testbench to an external SOC via the SOC interface

and the model interface. Therefore, as explained in greater detail below, Applicants

respectfully submit that the prior art of record does not teach or suggest the claimed

invention.

A. Independent claim 2

The Office Action argues that Blaner teaches a verification test bench system for

testing a system-on-a-chip (SOC) interface of an SOC, said verification test bench system

comprising a verification interface model connected to said SOC interface; and a test

bench external bus interface unit (EBIU) connected to said verification interface model,

wherein said test bench EBIU is connected to a SOC EBIU within said SOC, and wherein

10/060,750   13

said SOC EBIU allows a test case running in said SOC to control both said SOC interface

and said verification interface model (Office Action, pp. 3-4, sections 10-11). Such

features are defined in independent claim 2 using identical language.

More specifically, the Office Action argues that Blaner has the ability to connect a

memory-mapped external device, which contains software readable and writable registers

that appear as wires in a testbench, to an external bus (Office Action, p. 3, section 10

(citing Blaner, p. 208, column 1, para. 2)). However, nothing in Blaner mentions

connecting the testbench to an external SOC via the model interface and the SOC

interface. The "memory-mapped external device" of Blaner is not synonymous with the

"model interface" of Applicants' invention. Specifically, in Blaner, the "memory-

mapped external device" is used to synchronize external activity to internal software.

To the contrary, as described in paragraph 23 of Applicants' disclosure, the

invention transfers data from the external interface model 210 to the SOC interface 101.

The test case calls the software driver (SWD) 135-137 for the interface 101 and instructs

the software driver to configure the interface 101 to receive data. Next, the test case calls

the same SWD 135-137 and instructs the software driver to configure the external

interface model 210 to send data. The SOC's interface 101 and the external interface

model 210 are implemented to respond to different unique addresses. Thus, when the test

case calls the SWD 135-137 to perform some configuration on one of the interfaces, the

test case sends the address of that interface along with the operation to be performed. The

test case then sends test data to the unique data address of the external interface model

210. This data is sent from the SOC 100 through the SOC's external bus interface unit

10/060,750   14

(EBIU) 205 to the external EBIU 200 and then along to the interface model 210. From there the data is sent through the interface model 210 into the SOC's interface 101 which is configured to receive data. Once the data is back in the SOC 100, the test case checks it for correctness and a test status is recorded.

In addition, the Office Action argues that Blaner has the ability to use an EBIU to control up to eight banks of mixed types of memories and to operate at one-half the PLB clock frequency (Office Action, pp. 3-4, section 10 (citing p. 205, column 2, para. 3)). Further, the Office Action asserts that Blaner explains that the EBIU allows an off-chip device, called the external bus master, to take ownership of the external bus and access attached memories. However, nothing in Blaner mentions a test case in the SOC that can use the same software driver to program both the SOC interface and the model interface. As described in paragraph 24 of Applicants' disclosure, the invention **allows the test case executing in the SOC 100 to use the same software driver, if appropriate, to program both interfaces 101, 210** (to configure and control the SOC interface and the verification interface model). The test case can also use one driver to program the SOC interface 101 and another to program the interface model 210, both of which are controlled by the test case (a test case running in the SOC to control both the SOC interface and the verification interface model). This is an improvement over the conventional situation where the test case running within the SOC 100 controls the SOC interface 101, and another software program (written in a bus functional language) controls the external interface 210. The invention provides increased reusability and decreased development time because the invention uses the same or similar software

10/060,750   15

written in the same language to program both interfaces (the same software driver to configure and control the SOC interface and the verification interface model).

Furthermore, as described in paragraph 26 of Applicants' disclosure, the invention represents a clean way of controlling external interfaces without the need for a complex control mechanism such as the conventional semaphore derived scheme used to enable communication between the SOC being tested and the external interface. The external bus mastering of the test bench EBIU 200 allows external model programming from the SOC test case. **Thus, the same test case directly controls operations of the SOC interface 101 and the external model 210** (a test case running in the SOC to control both the SOC interface and the verification interface model).

Nothing in Blaner teaches the foregoing features of Applicants' invention, namely a test case in the SOC that can use the same software driver to program both the SOC interface and the model interface. The fact that Blaner discloses, on page 205, column 2, paragraph 3, an external bus master for taking ownership of the external bus and accessing attached memories has nothing to do with programming both the SOC interface and the model interface with the same software driver.

Blaner broadly discloses a design for an SOC and briefly discusses how it was tested. Blaner includes many details of how code was structured for re-useablility across the many SOC designs and how simulation is accelerated by modeling the processor in the RTX code, rather than using the HDL model of the processor. However, Blaner specifically does not disclose building a testbench to stimulate/monitor the chip's I/O.

10/060,750   16

With respect to Devins, the Office Action proposes that Devins has the ability to couple external bus interface logic of a device to the SOC device via a chip-external bus. However, nothing in Devins mentions connecting the testbench to an external SOC via the model interface and the SOC interface. The "external bus interface logic" of Devins is not synonymous with the "model interface" of Applicants' invention; rather, as pointed out on pages 12-13, item 28 of the Office Action, "[t]his corresponds to the 'test bench external bus interface unit (EBIU)'".

The Office Action also relies upon Devins' external bus interface logic 202 to reject Applicants' claims towards (1) a test case that utilizes the same software driver to configure and control the SOC interface and the verification interface model (i.e., dependent claims 11, 17, 24, and 31); and, (2) an SOC EBIU that allows a test case running in the SOC to control both the SOC interface and the verification interface model (i.e., independent claim 2 and dependent claims 9, 16, 22, and 29). Specifically, the Office Action highlights Devins' external bus interface logic 202, which is designed to direct signals received via connection 107 to the appropriate logical address, and to convert the particular bus protocol received into an internally-used format applicable to the command decode logic 203 (column 4, lines 15-20 and 38-40; column 5, lines 5-8).

Once more, the features cited in the prior art reference have nothing to do with utilizing the same software driver to configure and control the SOC interface and the verification interface model. Further, Devins does not mention allowing a test case running in the SOC to control both the SOC interface and the verification interface

10/060,750   17

model; nor does Devins mention programming both the SOC interface and the verification interface model by the test case running in the SOC.

In addition, the Office Action argues that Blaner expressly teaches connecting the testbench to an external SOC via the SOC interface and the model interface. Such a feature is defined in independent claim 2 using the following language: "[a] verification test bench system for testing a system-on-a-chip (SOC) interface of an SOC, said verification test bench system comprising: a verification interface model connected to said SOC interface".

Specifically, the Office Action asserts that the test operating system (TOS) is capable of scheduling test programs for execution and multi-tasking operations, enabling concurrent core execution. This provides the appropriate intercore and bus transactions required for exercising the chip (Office Action, pp. 17-18, section 45(c) (citing Blaner, p. 207, Section IV "Design Verification")). The section of Blaner cited by the Office Action describes the function and performance of the testbench system – it does not disclose the structural elements of an SOC interface and a model interface, which connects the testbench to an external SOC.

The Office Action also asserts that Devins expressly teaches an external memory-mapped test device (EMMTD) that is coupled between a SOC design being tested in simulation, and cores external to the SOC design. Further, the Office Action argues that Devins discloses that a test case being executed for SOC verification by a simulated embedded processor in the SOC can communicate with and control elements external to the SOC, by using the EMMTD to perform such functions as initiating external core logic

10/060,750   18

which drives test signals to an internal core, directly controlling an internal core via its external interface, or determining the status of an external core. Although Devins broadly discusses an SOC coupled to external components, Devins does not teach or suggest connecting the testbench to an external SOC via the SOC interface and the model interface.

More specifically, as illustrated in FIG. 1 of Applicants' invention, the SOC 100 is connected to the verification test bench 300. This connection is made by connecting the SOC interface 101 (of the SOC 100) and the verification interface model 210 (of the verification test bench 300). Moreover, as described in paragraph 25 of Applicants' disclosure, "[i]n FIG. 1, the invention transfers data from the external interface model 210 to the SOC interface 101 ... the data is sent through the interface model 210 into the SOC's interface 101 which is configured to receive data. Once the data is back in the SOC 100, the test case checks it for correctness and a test status is recorded." Thus, Applicants respectfully submit that, unlike the claimed invention, neither Blaner nor Devins teaches connecting the testbench to an external SOC via the SOC interface and the model interface, as defined in independent claim 2.

B.   Independent claim 8

The Office Action also argues that Blaner teaches a verification test bench system for testing a system-on-a-chip (SOC) interface of an SOC, said verification test bench system comprising a verification interface model connected to said SOC interface; and a test bench external bus interface unit (EBIU) connected to said verification interface

10/060,750   19

model, wherein said test bench EBIU is connected to a SOC EBIU within said SOC, and

wherein said test bench EBIU and said SOC EBIU are mastered by the same processor in

said SOC (Office Action, pp. 6-7, section 17). Such features are defined in independent

claim 8 using identical language.

As described more fully above, neither Blaner nor Devins teach or suggest a

verification test bench system for testing a system-on-a-chip (SOC) interface of an SOC,

said verification test bench system comprising a verification interface model connected to

said SOC interface; and a test bench external bus interface unit (EBIU) connected to said

verification interface model, wherein said test bench EBIU is connected to a SOC EBIU

within said SOC.

The Office Action asserts that Blaner teaches that wrap backs are utilized as much

as possible, and behaviorals often contain hard-coded packet or stream data (Office

Action, p. 7, section 17 (citing Blaner, p.208)). As also noted in the Office Action, the

Examiner interprets that "wrap backs" do not containing functioning processors, and

therefore must rely on the SOC processor. Applicants respectfully submit that even if

wrap backs do not contain functioning processors, it cannot be assumed that the wrap

backs are mastered by the same processor. Moreover, Blaner does not teach that the

wrap backs are mastered by a processor in the SOC as opposed to the verification

testbench. In addition, the wrap backs disclosed in Blaner are not analogous to the test

bench EBIU and the SOC EBIU of the claimed invention. Specifically, a first wrap back

is not connected to a verification interface model and a second wrap back, wherein the

second wrap back is in the SOC. Therefore, it is Applicants' position that Blaner does

10/060,750   20

not teach or suggest a test bench EBIU and an SOC EBIU that are mastered by the same processor in said SOC, as defined in independent claim 8.

C.   Independent claim 15

The Office Action argues that Blaner teaches a verification test bench system for testing a system-on-a-chip (SOC) interface of an SOC, said verification test bench system comprising a verification interface model connected to said SOC interface; and a test bench external bus interface unit (EBIU) connected to said verification interface model, wherein said test bench EBIU is connected to a SOC EBIU within said SOC, and wherein said test bench EBIU and said SOC EBIU are mastered by the same processor in said SOC, such that said SOC interface and said verification interface model are programmed by the same test case running in said SOC. (Office Action, pp. 7-9, section 18). Such features are defined in independent claim 15 using identical language.

As described more fully above, neither Blaner nor Devins teach or suggest a verification test bench system for testing a system-on-a-chip (SOC) interface of an SOC, said verification test bench system comprising a verification interface model connected to said SOC interface; and a test bench external bus interface unit (EBIU) connected to said verification interface model, wherein said test bench EBIU is connected to a SOC EBIU within said SOC, and wherein said test bench EBIU and said SOC EBIU are mastered by the same processor in said SOC.

The Office Action relies on the EBIU in Blaner to reject Applicants' claims directed towards the SOC interface and the verification interface model, which are both

10/060,750   21

programmed by a test case running in the SOC (i.e., independent claim 15 and dependent

claims 10, 23, and 30). Specifically, the Office Action argues that Blaner has the ability

to use the EBIU to: (1) control up to eight banks of mixed types of memories; (2) operate

at one-half the PLB clock frequency; and, (3) allow the external bus master to take

ownership of the external bus and access attached memories (Office Action, p. 8, section

18 (citing Blaner, p. 205, column 2, para. 3)). Again, the Office Action argues that

Blaner discloses, on page 205, column 2, paragraph 3, an external bus master for taking

ownership of the external bus and accessing attached memories, which has nothing to do

with an SOC EBIU that allows a test case running in the SOC to control both the SOC

interface and the verification interface model; and, the SOC interface and the verification

interface model, which are both programmed by a test case running in the SOC.

Blaner broadly discloses a design for an SOC and briefly discusses how it was

tested. Blaner includes many details of how code was structured for re-useablility across

the many SOC designs and how simulation is accelerated by modeling the processor in

the RTX code, rather than using the HDL model of the processor. However, Blaner

specifically does not disclose building a testbench to stimulate/monitor the chip's I/O.

The Office Action also relies upon Devins' external bus interface logic 202 to

reject Applicants' claims towards the SOC interface and the verification interface model,

which are programmed by the test case running in the SOC (i.e., independent claim 15

and dependent claims 10, 23, and 30). Specifically, the Office Action highlights Devins'

external bus interface logic 202, which is designed to direct signals received via

connection 107 to the appropriate logical address, and to convert the particular bus

10/060,750   22

protocol received into an internally-used format applicable to the command decode logic
203 (column 4, lines 15-20 and 38-40; column 5, lines 5-8).

Once more, the features cited in the prior art reference have nothing to do with
utilizing the same software driver to configure and control the SOC interface and the
verification interface model. Further, Devins does not mention allowing a test case
running in the SOC to control both the SOC interface and the verification interface
model; nor does Devins mention programming both the SOC interface and the
verification interface model by the same test case running in the SOC, as defined in
independent claim 15.

D. Independent claims 21 and 28

The Office Action argues that in regards to independent claims 21 and 28, Blaner
teaches connecting a memory-mapped external device, which contains software readable
and writable registers that appear as wires in a testbench, to an external bus (Office
Action, p. 9, section 19 (citing Blaner, p. 208)). However, nothing in Blaner mentions
connecting the testbench to an external SOC via the model interface and the SOC
interface. The "memory-mapped external device" of Blaner is not synonymous with the
"model interface" of Applicants' invention. Specifically, in Blaner, the "memory-
mapped external device" is used to synchronize external activity to internal software.

To the contrary, as described in paragraph 23 of Applicants' disclosure, the
invention transfers data from the external interface model 210 to the SOC interface 101.
The test case calls the software driver (SWD) 135-137 for the interface 101 and instructs

10/060,750   23

the software driver to configure the interface 101 to receive data. Next, the test case calls

the same SWD 135-137 and instructs the software driver to configure the external

interface model 210 to send data. The SOC's interface 101 and the external interface

model 210 are implemented to respond to different unique addresses. Thus, when the test

case calls the SWD 135-137 to perform some configuration on one of the interfaces, the

test case sends the address of that interface along with the operation to be performed. The

test case then sends test data to the unique data address of the external interface model

210. This data is sent from the SOC 100 through the SOC's external bus interface unit

(EBIU) 205 to the external EBIU 200 and then along to the interface model 210. From

there the data is sent through the interface model 210 into the SOC's interface 101 which

is configured to receive data. Once the data is back in the SOC 100, the test case checks it

for correctness and a test status is recorded.

In addition, the Office Action argues that Blaner has the ability to use an EBIU to

control up to eight banks of mixed types of memories and to operate at one-half the PLB

clock frequency (Office Action, p. 9, section 19 (citing Blaner, p. 205, section II)).

Further, the Office Action asserts that Blaner explains that the EBIU allows an off-chip

device, called the external bus master, to take ownership of the external bus and access

attached memories (Office Action, p. 9, section 19 (citing Blaner, p. 205, section II)).

However, nothing in Blaner mentions a test case in the SOC that can use the same

software driver to program both the SOC interface and the model interface. As described

in paragraph 24 of Applicants' disclosure, the invention allows the test case executing in

the SOC 100 to use the same software driver, if appropriate, to program both

10/060,750   24

interfaces **101, 210** (to configure and control the SOC interface and the verification interface model). The test case can also use one driver to program the SOC interface 101 and another to program the interface model 210, both of which are controlled by the test case (a test case running in the SOC to control both the SOC interface and the verification interface model). This is an improvement over the conventional situation where the test case running within the SOC 100 controls the SOC interface 101, and another software program (written in a bus functional language) controls the external interface 210. The invention provides increased reusability and decreased development time because the invention uses the same or similar software written in the same language to program both interfaces (the same software driver to configure and control the SOC interface and the verification interface model).

Furthermore, as described in paragraph 26 of Applicants' disclosure, the invention represents a clean way of controlling external interfaces without the need for a complex control mechanism such as the conventional semaphore derived scheme used to enable communication between the SOC being tested and the external interface. The external bus mastering of the test bench EBIU 200 allows external model programming from the SOC test case. Thus, **the same test case directly controls operations of the SOC interface 101 and the external model 210** (a test case running in the SOC to control both the SOC interface and the verification interface model).

Nothing in Blaner teaches the foregoing features of Applicants' invention, namely a test case in the SOC that can use the same software driver to program both the SOC interface and the model interface. The fact that Blaner discloses, on page 205, column 2,

10/060,750   25

paragraph 3, an external bus master for taking ownership of the external bus and accessing attached memories has nothing to do with programming both the SOC interface and the model interface with the same software driver.

Therefore, contrary to the position taken in the Office Action, Applicants submit that Blaner does not teach or suggest the method according to claim 21, and the program storage device according to claim 28, comprising: connecting a verification interface model to the SOC interface; connecting a test bench external bus interface unit (EBIU) to the verification interface model; connecting the test bench EBIU to a SOC EBIU within the SOC; and comparing the SOC interface with the interface model.

II.   Formal Matters and Conclusion

Therefore, it is Applicants' position that neither Blaner nor Devins teach or suggest many features defined by independent claims 2, 8, 15, 21, and 28 and that such claims are patentable over the prior art of record. Further, it is Applicants' position that dependent claims 9-14, 16-20, 22-27, and 29-34 are similarly patentable, not only because of their dependency from patentable independent claims, but also because of the additional features of the invention they defined.

In view of the foregoing, the Examiner is respectfully requested to reconsider and withdraw the rejections to the claims. Accordingly, Applicants submit that claims 2, and 8-34 are patentably distinct from the prior art of record and are in condition for allowance. The Examiner is respectfully requested to pass the above application to issue at the earliest possible time.
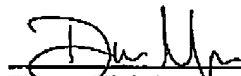
10/060,750   26

Should the Examiner find the application to be other than in condition for

allowance, the Examiner is requested to contact the undersigned at the local telephone

number listed below to discuss any other changes deemed necessary. Please charge any

deficiencies and credit any overpayments to Attorney's Deposit Account Number 09-

0456.

Respectfully submitted,

Dated: 3|30|06

Duane N. Moore
Registration No. 53,352

Gibb I.P. Law Firm, LLC
2568-A Riva Road, Suite 304
Annapolis, MD 21401
Voice: (410) 573-6501
Fax: (301) 261-8825
Customer Number: 29154

10/060,750 27